REQUIREMENTS OF USEFUL REMOTE LABS

Javier Garcia-Zubia, Iván Trueba, Diego Lopez-de-Ipiña, Pablo Orduña, Unai Hernandez, Jose M. Saenz

Faculty of Engineering, University of Deusto, Avd. Universidades 24, 48007 Bilbao, Spain, zubia@eside.deusto.es

Abstract: The Faculty of Engineering of the University of Deusto has had available a WebLab oriented to Microelectronics since 2001. Currently, the field of WebLab design is very active, and several other universities are adopting them as a sign of quality and distinction for Microelectronics teaching, as we have done. The first part of this paper shows several requirements that a WebLab should meet and the next points presents WebLab-Deusto.

Keywords: Remote Labs

1. KINDS OF REMOTE LABORATORIES

Currently, the European university framework is changing as a result of the Bologna Declaration. Thus, the universities are trying to decentralize their activities making the classrooms and laboratories available anywhere and at anytime. Accordingly, WebLabs or Remote Labs are growing in importance and many faculties and laboratories are developing this technology to enhance their educational offer.

WebLabs may be classified in three areas according to the type of control established:

1. **Remote instrumentation**: WebLab consists of one or more experiments where users can only activate their inputs (virtual switches, signal generators,…) and see their virtual or real outputs through a webcam (LEDs, signals in a oscilloscope,…) One example of this WebLab is the Remote Access Laboratory (www.ral.ul.ie) of University of Limerick and the Remote Lab of Blekinge Institute of Technology (Gustavsson et al, 2005) (see Fig. 1), where the user can test a PLL changing the VCO control voltage, for example, and then measure PLL's output signal connected to a oscilloscope focused by a webcam.
2. **Remote parameter control**. The main difference between this WebLab type and the previous one is that here the user is able to change control parameters in order to modify the logic of the system. PID control is the most significant example of this type of WebLab, where normally the user cannot reconfigure the regulator's structure. One example of this WebLab is the Automatic Control Telelab of University of Siena (Casini et al, 2003) (see Fig. 2). In this WebLab, the user can manipulate some parameters (position control, speed control, level control, flow control …) that influence the control logic of a model and the results can then be watched using a webcam.
3. **Remote control logic**. In this case, the user can change both the logic and the system control parameters. A simple example would be a classic didactic model (LEDs, 7-segments,…) controlled by a CPLD or FPGA that has been loaded with the student program. The same could be said for a microcontroller, a DSP, a PLD or a PC-controlled system by a LabView program. Here, the risk is the destruction of the system due to a programming error because the student has the whole control over all the variables of experiment. Good examples of this type of WebLab are WebLab-DEUSTO (Gracia et al, 2005) (see Fig. 3), the Shell & Tube Heat Exchanger Experiment involved in iLabs of MIT (Alamo, 2001) or the Remote Laboratory Project at the Blekinge Institute of Technology (Sweden) (Gustavsson et al, 2005).

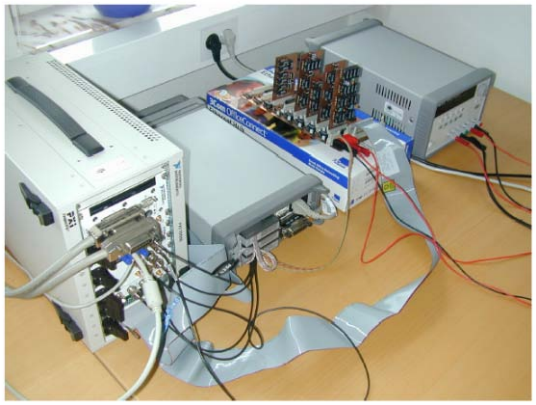


Fig. 1. Remote Instrumentaion of Blekinge Institute of Technology

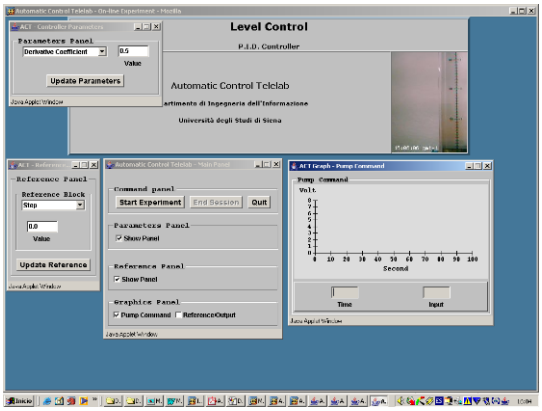
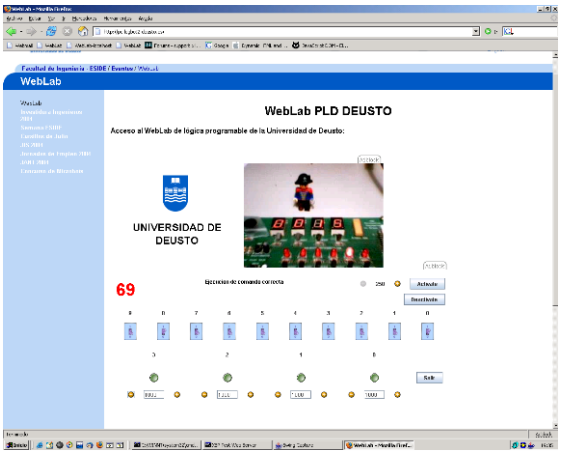


Fig. 2. Remote Parameter Control of University of Siena



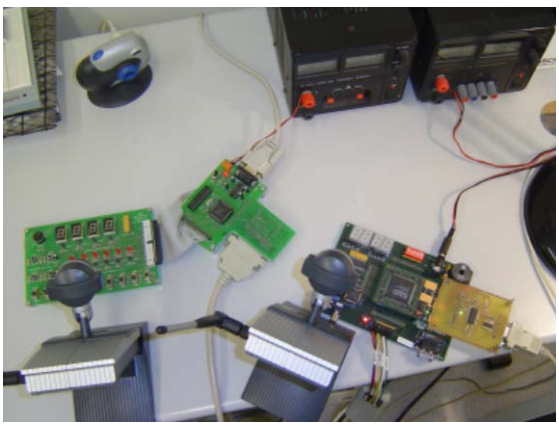


Fig. 3. Remote Control Logic of University of Deusto, web page and hardware.

2. REQUIREMENTS FOR USEFUL WEBLABS

The most powerful WebLab but not necessary the most complex, is the last one, because it includes all the previous ones. The real complexity of a Weblab depends more on didactic and service quality than on its type. The first question is very important, and many times is forgotten by some WebLab developers. However, some others authors, like Soysal (Soysal, 2000), Ponta (Ponta et al, 2005), Barron (Barron, 2005) and García-Zubía (Garcia-Zubia et al, 2005), remark its importance. The quality of service and the complexity of a WebLab depends on the following questions:

- **Is it didactic?** Is it inside of an educational platform? Does it satisfy objectives of the subject? Has it feedback from the students? Does it make student's work easier? Does it allow the collaborative learning? Does the student feel lack of control of the WebLab? Does the student only watch the WebLab or is he involved in it? Are reports and folders generated for student and teacher? Is the quality of the algorithms controlled in order to avoid bad use and the destruction of equipments? Is it easy and visual the control of inputs, the loading of programs, watching outputs…? Are there manuals, help…available?
- **Is it universal?** Is it operative on a 24x7 basis? Is it only accessed by teachers and some guests? Is it accessed by students of other universities? Is it available in several languages? Has the server other experiments, models, robots, etc. connected? Is it possible to be simultaneously accessed? Is it allowed remote experimentation, control and reconfiguration?
- **Is it professional?** Is the WebLab designed using advanced techniques included in web 2.0? Is it integrated into university IT? Are management and administration of the WebLab (passwords, e-mail, login…) automatic? Is the time management optimum? Does the server manage the users waiting to use the WebLab? How often is the server down? Would the IT services consider that the WebLab is according with the security policies? Do the WebLab maintainers keep quality of service polls regularly?
- **Is it technologically advanced?** Does the WebLab support mobile device access? Does it use microservers? Is it fully dependent on the server? Does it use XML? Does it use semantic technologies? How is the communication with the controlled devices (CPLD, oscilloscopes): RS232, USB, Ethernet…? Does the WebLab depend on previously installed software on the client side: jvm, Internet Exporer, Mozilla, Macromedia Flash…? Is it multiplatform: Linux, Windows…? Is the quality of capturated images good? Does the WebLab rely on proprietary software: LabView, etc.? Is it freeware, GNU GPL, copyleft? Has the WebLab a user friendly GUI? Is there a weblab-Google?

This list only tries to show different features of WebLabs. Each WebLab developer should try to answer those questions and make new ones.

3. WEBLAB-DEUSTO

WebLab-Deusto is a Remote Lab focuses on digital electronics, specially in CPLD and FPGA, but now is incorporating microcontrollers and DSP . The current version has been designed using web 2.0 technologies (see Fig. 4). A single client application shown in the user's browser communicates with the server through HTTP. We now have a web-based firewall-safe system programmed with AJAX (Asynchronous JavaScript and XML – http://en.wikipedia.org/wiki/AJAX). The main benefit of AJAX is that it works on any web browser, without any plug-in installation required. The client application is now a pure HTML/JavaScript solution which follows the AJAX web interaction model. This technology is being applied successfully to sophisticated web applications such a Gmail, Google Maps or Flick, which have been termed as belonging to next web generation, i.e. Web 2.0. The server side is composed of the elements of the previous version plus a new ASP.NET application, based on Mono, offering a SOAP Web Service interface to client applications.

In this AJAX-based web solution, the WebLab server may be run both in Microsoft Windows and GNU/Linux, and it does not rely on Java anymore. We have also managed to run the client application under Nokia mobile devices running Symbian OS and equipped with an Opera browser.

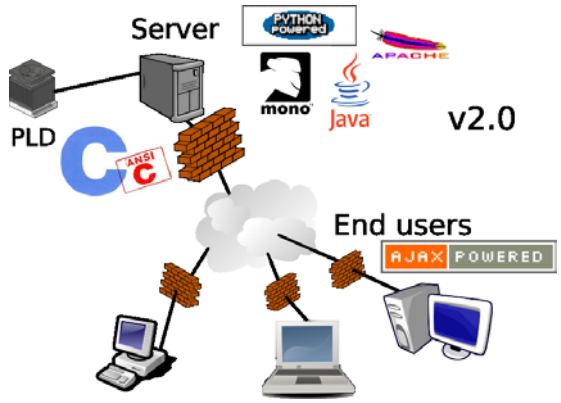


Fig. 4. WebLab-Deusto 2.0

3.1 Microserver-Based AJAX Web Solution

We are currently progressing towards the WebLab architecture shown in Fig. 5. This solution will be web-based, firewall-safe and more scalable (will provide several IP-accessible programmable devices). Many groups of users from any client platform will be able to access simultaneously any of the several networked programmable devices, allowing collaborative learning.

This fourth generation is again a multi-platform solution, which supports both Windows and GNU/Linux on the server-side. Moreover, there is only one programming language used: Python, maintaining the security features of the previous iterations. However, the most outstanding contribution of this iteration is the incorporation of microservers. A microserver adds to the WebLab hardware an IP address and network-based programmability. All the communication between the server and the WebLab board, previously undertaken by means of RS232 and the PIC, is now undertaken through the Internet.

The adoption of microservers opens many possibilities: use of XML (the de-facto language for data exchange), creation of autonomous WebLabs without the need of a centralized server, makes feasible the creation of intranet hardware networks and so on. Besides, the microservers are versatile, powerful and low cost (around € 100) hardware. Anyhow, the main drawback of adopting microservers is based on being a recent technology where very basic, non-sophisticated services have been deployed so far. Furthermore, it produces a dramatic change to the traditional client/server-based WebLab architecture, which can now move into a more decentralized P2P architecture. Thus, many of the functional blocks currently allocated to the server can be moved to the microservers.

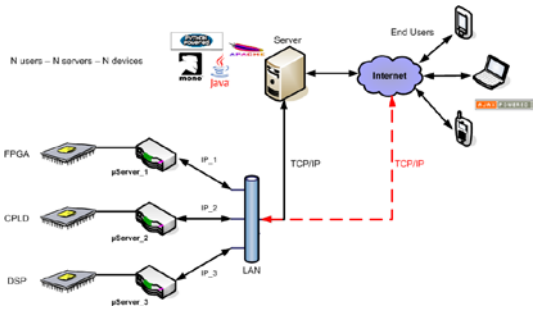


Fig. 5. WebLab-Deusto 3.0

4. CONCLUSIONS AND FUTURE WORK

This work has provided three main contributions. Firstly, it is very important to pay extra attention to the software-side of a WebLab design, even more than to the hardware-side, since many problems of deployed WebLabs come from poor software-side designs (accessibility, security, and so on). Secondly, the use of microservers on the hardware-side will revolutionise and encourage the usage and design of WebLabs. Thirdly, academically it is obvious that the use of a WebLab improves the subject teaching and the opinion that students have about the labs, the subjects and the lecturers. Anyhow, it is always important to control the quality of new developments in a WebLab, checking the students' opinion. Currently, our research group is working in three aspects: a) extending the use of WebLab-DEUSTO to microcontrollers and DSP, b) redesigning WebLab-DEUSTO by adopting microservers and c) documenting the academic performance of WebLab-DEUSTO.

REFERENCES

Alamo, J.A., (2001) MIT Microelectronics Weblab, Marzo, 27. http://web.mit.edu

Barrón, M. (2005) “Laboratorios virtuales para enseñanza por internet" en *I Jornadas de Tendencias sobre eLearning, TEL 2005*, Madrid (Spain).

Casini, M.; Prattichizzo, D. y Vicino, A. (2003) “e-Learning by Remote Laboratories: a new tool for control education” . The 6th IFAC Conference on Advances in Control Education, Finland.

García Zubía, J. (2004) “Programmable Logic and WebLab”, *Proceedings of the 5th European Workshop on Microelectronics Education*, Lausanne (Switzerland), ISBN: 1-4020-2072-4, PP: 277-282.

García-Zubia, J et al. (2005) “A new approach for implementing remote laboratories: a practical case”. *2nd International Symposium REV 2005*, Brasov (Romania), ISBN: 3-89958-137-7.

Gustavsson, I. et al. (2005) “A flexible remote electronics laboratory". *2nd International Symposium in Remote Engineering and Virtual Instrumentaion, REV 2005*, Brasov (Romania).

Ponta, D. et al (2005) "Remote experiments in electronics: pedagogical issues" *2nd International Symposium REV 2005*, Brasov (Romania), ISBN: 3-89958-137-7.

Soysal, O. (2000) "Computer Integrated Experimentation in Electrical Engineering Education over Distance" *Proceedings of ASEE 2000 Annual Conference*, Saint Louis, MO.